

36628
SPIRAS
PROGRAMMER'S
CALENDAR ©



Spiras Systems, Inc.
Affiliate of
USM Corporation

RANDOM NUMBER GENERATORS

(1) $R' = (2^P + 1)R + C$

R' = NEW RANDOM FRACTION.
R = PREVIOUS RANDOM FRACTION.
P = 7 TO 21 (BEST WHEN = 9).
C = ANY ODD NO. (BEST WHEN = 1).
N = MAX. RANGE OF RANDOM INTEGER.

(2) $\text{RANDOM INTEGER} = R'(N)$

TYPICAL PROGRAM

```
LDA R
ALS 7
ADD R
ADD =1
STO R
MPY N
JRT EXIT
```

↑
NO. OF
TIMES EACH
VALUE OR
R# IS
GENERATED

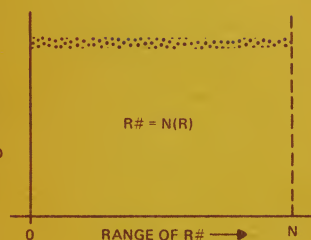


FIG. 1

FIG. 2

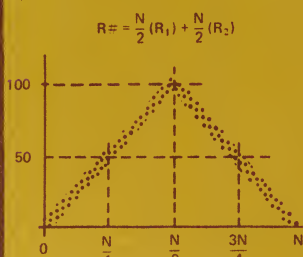


FIG. 3

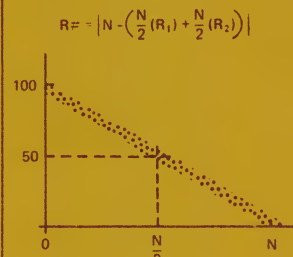


FIG. 4

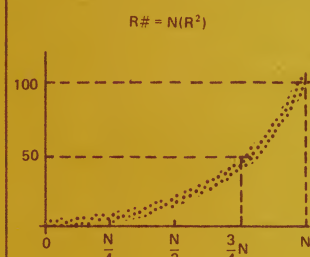


FIG. 5

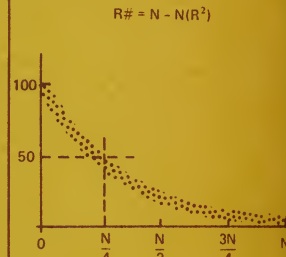
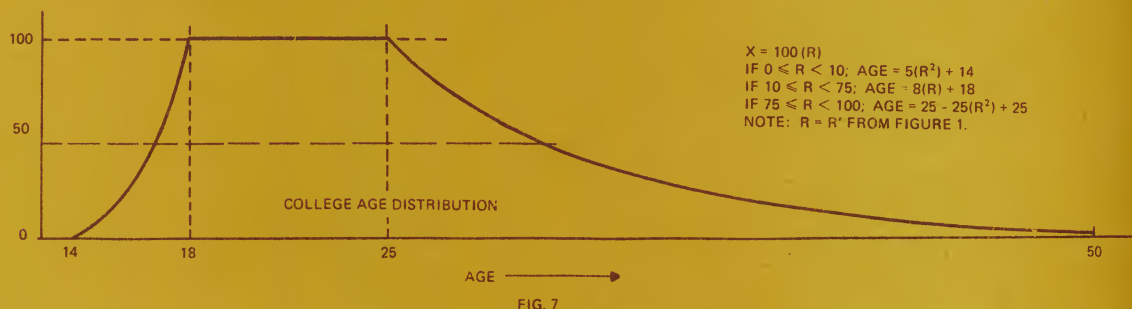


FIG. 6



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

JANUARY							3 6 6 2 8
SUN	MON	TUES	WED	THUR	FRI	SAT	
				0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	
0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	
0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	
1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	
1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1	

Random number generators are useful in many computer applications including simulated sampling or for demonstrations. The procedure discussed in the following paragraphs will be used for decision-making in the game demonstrations that will be presented in later months.

A very simple random number generator was presented in the Journal of ACM (January 60, Page 75) and has the advantage of simple coding, long recycle period ($=2^N$ where N = no. of bits in word), and insensitivity to the starting value (could even be zero). Figure 1 gives the formula and a sample program for its implementation.

If the string of bits in the output number generated by this generator is thought of as a fraction with binary point at the extreme left, then the random number is between 0 and 1 (but never equal to 1). Now, if this fraction was multiplied by an integer N with its binary point to the extreme right, the result would be a random integer with its binary point to the extreme right and between 0 and N (but never equal to N).

If many random numbers were generated between 0 and N , using the formula in Figure 1, the frequency distribution of R between 0 and N is a straight line as shown in Figure 2.

Figure 3 shows the frequency distribution of R when two random numbers are generated between 0 and $N/2$ and then these two numbers are added to give a random number (R) between 0 and N . If this value of R is then subtracted from $N/2$ and the result made positive, this new value of R would have a frequency distribution curve as shown in Figure 4.

If the random fraction between 0 and 1, resulting from Figure 1, is multiplied by itself before multiplying by N , the resulting frequency distribution of R is an exponential curve as shown in Figure 4. A left exponential distribution curve as shown in Figure 5 is generated if R is subtracted from N .

As an example of how these random number generators can be used, we wish to randomly pick the age of a college student. We are told that 10% are under 18, 25% are over 25, and the rest of the students are between the ages of 18 and 25. The distribution of ages required is shown in Figure 7. Our random number generator, then, has the following steps:

- (1) Generate a random number with $N = 100$ as shown in Figure 2. If this number is below 10, go to step 3. If this number is below 35, go to step 4. Otherwise continue with step 2.
- (2) Generate a random number with $N = 8$ as shown in Figure 2. Add 18 to this number and exit with the age.
- (3) Generate a random number with $N = 5$ as shown in Figure 5. Add 14 to this number and exit with the age.
- (4) Generate a random number with $N = 25$ as shown in Figure 6. Add 25 to this number and exit with the age.

Using these random number generators and your imagination, many interesting distributions and uses will quickly result.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65 K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

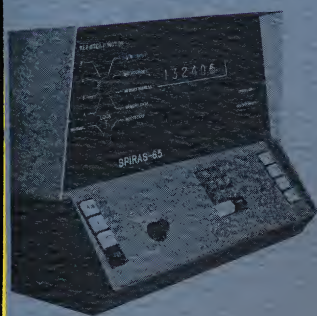
SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



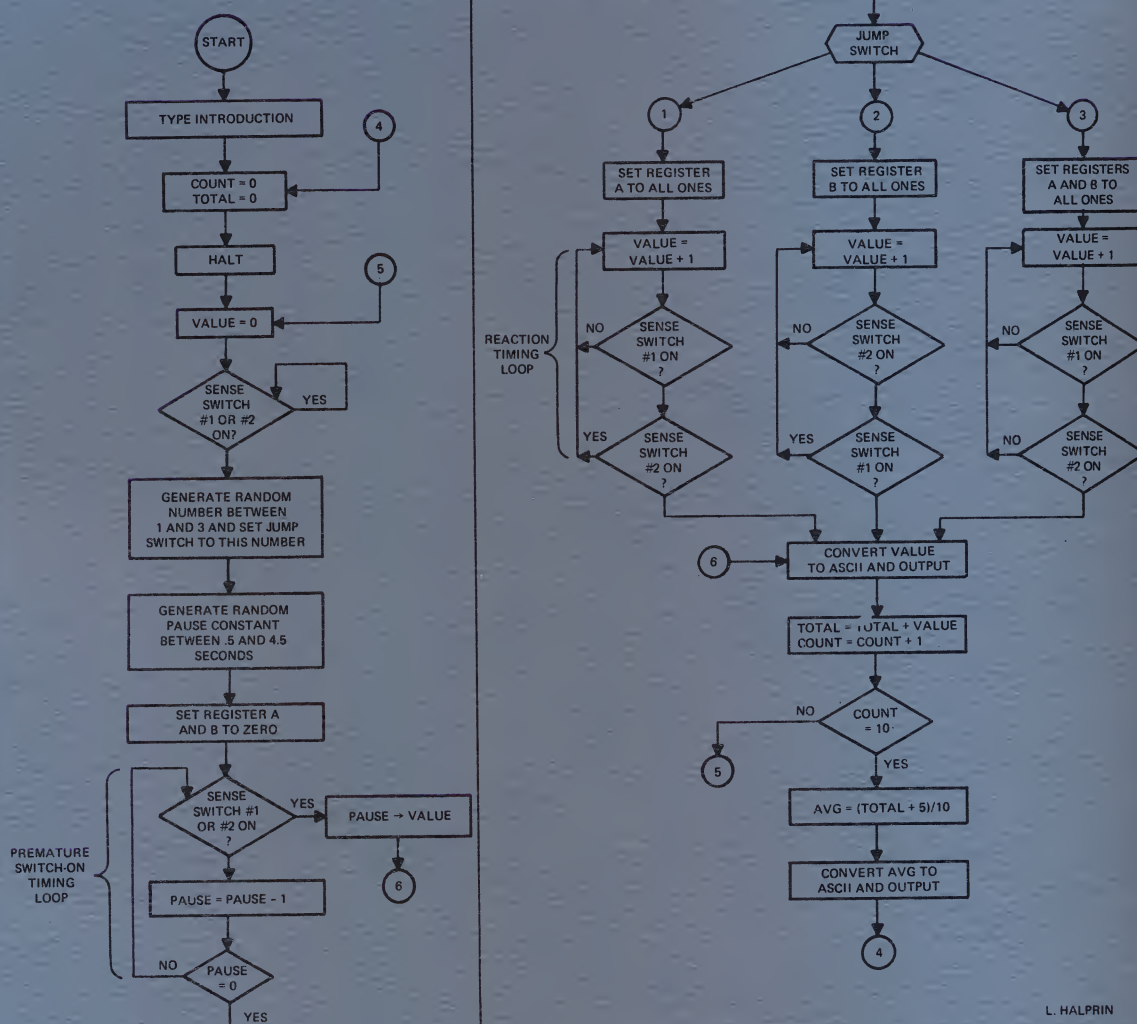
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



REACTION DEMONSTRATION



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR[©]

FEBRUARY					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1
0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0
0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1
1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0

It is well known that the primary advantage the computer has over humans is its superior speed. Yet it is of great interest to most persons just how well their "intelligent" reaction compares with the computer's "dumb" reaction.

This demonstration program allows a person's reaction time to a set of 3 displays to be measured and compared with reaction times generated by other persons. The computer is presumed to have at least two programmable register displays and two sense switches on its control console. After a random length pause, one or both of the registers will be flashed on (set to all ones). The choice of register 1, 2, or both, is also randomly made. As soon as the register(s) is flashed on, the operator is to turn on sense switch 1, 2, or both. The time between the flash-on point to the turn-on point is the operator's reaction time.

Ten measurements are made on each operator and the average reaction time is computed for competition purposes with other operators. Two variables help make the game interesting. The first variable is the random length pause before the registers are flashed, making it hard for the operator to anticipate the flash-on point. The second variable is not knowing which of three combinations of the registers will be flashed - each requiring a different operator response. A check is also made to penalize the operator if his response occurs before the flash-on point.

The reaction time given by the computer is a measurement of time but not in units of seconds. What units the number represents is determined by the number of instructions in the timing loop and the execution time of the computer. A conversion factor could be determined in order to give the reaction time in microseconds. The timing loops should be adjusted so the operator will be penalized perhaps 100% for a switch turn-on before the register flash-on. For instance, if the operator turned on a switch 50 msec before flash-on time, his reaction time would be considered as 100 msec. This penalty discourages premature switch turn-on which might otherwise be prevalent with equalized timing loops before and after switch turn-on.

Almost any two programmable indicators can be used as a flash-on indicator for this demonstration, such as sense lights, or the overflow and improper divide error indicators; or the console typewriter might be used to type the digits 1, 2, or 3. Reaction input to the computer is best done with sense switches but any input which can be made while the computer is in dynamic mode is acceptable, such as typewriter input (if the computer can compute during input) or switches connected to the computer's trapping or interrupt circuits.

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



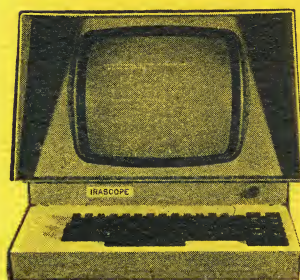
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



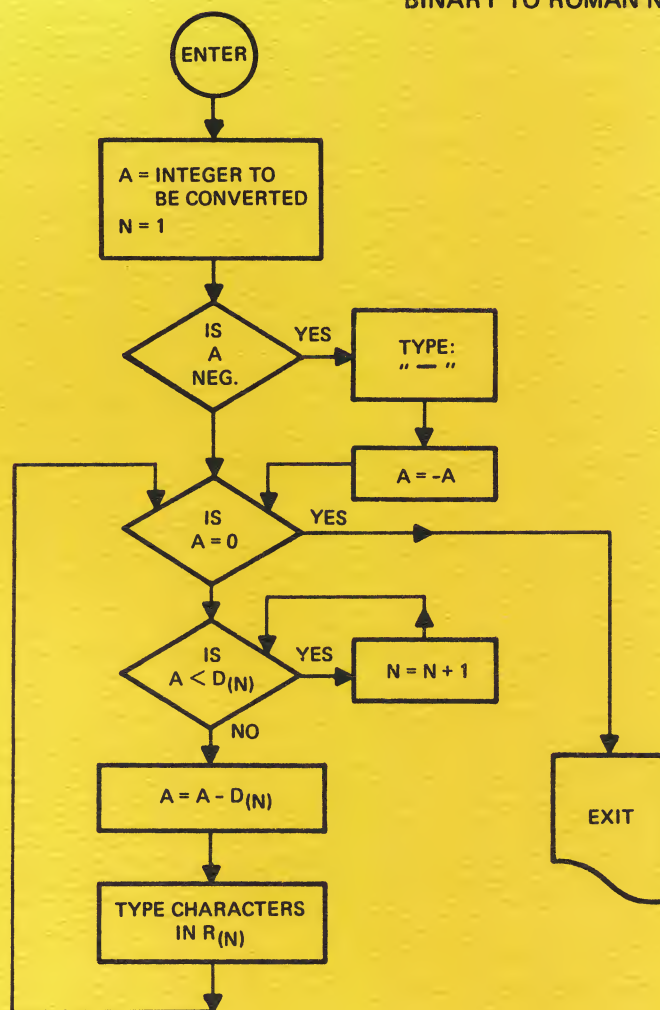
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



BINARY TO ROMAN NUMERAL CONVERSION



REDUCTION TABLE		
N	D	R
1	1,000,000	M/
2	900,000	C/M/
3	500,000	D/
4	400,000	C/D/
5	100,000	C/
6	90,000	X/C/
7	50,000	L/
8	40,000	X/L/
9	10,000	X/
10	9,000	MX/
11	5,000	V/
12	4,000	MV/
13	1,000	M
14	900	CM
15	500	D
16	400	CD
17	100	C
18	90	XC
19	50	L
20	40	XL
21	10	X
22	9	IX
23	5	V
24	4	IV
25	1	I

L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

MARCH				3 6 6 2 8		
SUN	MON	TUES	WED	THUR	FRI	SAT
0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1
0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0
0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1
1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0
1 1 1 0 1	1 1 1 1 0	1 1 1 1 1				

Conversion of numbers from binary to decimal is a common function used by almost every computer program. The decimal numbers are expressed by a combination of Arabic numerals 0 through 9. This, however, is not the only way to express numbers. In fact, long before the Arabic-decimal system of number notation became popular, the Romans had their own number system called, surprisingly enough, Roman Numerals.

Since many computer manufacturers and software houses are expanding their marketing area to foreign countries, it may become necessary to convert a binary number to a Roman Numeral format (in Rome or Romania perhaps?). Conversion of binary numbers to Roman Numerals does not lend itself to a mathematical conversion due to lack of symmetrical relationship between the "digits" of the Roman Numeral system. A reduction conversion method is used instead, as shown in the flowchart.

The Romans had no concept of fractions and so all their numerals were integers. At the time their system was initiated, one thousand was as large as could be conceived and was given the letter M (magna = large). Other decimal equivalents are shown in Figure 1. A later modification to the Roman Numeral system extended the range a thousand fold by defining any "digit" with a Bar over it as being 1000 times the normal value. Thus, $\overline{M} = 1,000,000$. Since most computer typewriters or printers have no over-bar, the bar is instead placed on the right of the character, thus $M = 1000$, $M/ = 1,000,000$. In this way, numbers as high as one million can be expressed and by use of multiple $M/$'s, even larger numbers are converted. If all arithmetic is done in integer format, the conversion is exact.

Example: 3398353 = M/M/M/C/C/C/X/C/V/MMMCCCLIII
1964 = MCMLXIV

M/ = 1,000,000	L/ = 50,000	M = 1,000	L = 50
D/ = 5,000,000	X/ = 10,000	D = 500	X = 10
C/ = 100,000	V/ = 5,000	C = 100	V = 5
Figure 1: Roman/Decimal Equivalents			I = 1

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.

FOR

Industrial/Manufacturing Processes
Numerical Machine Tools
Data Acquisition and Analysis
Automatic Test and Instrumentation
Function Generation
CRT Display Graphics Generation
Communications Data Concentration and Distribution
Scientific Data Processing
Business Data Processing

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



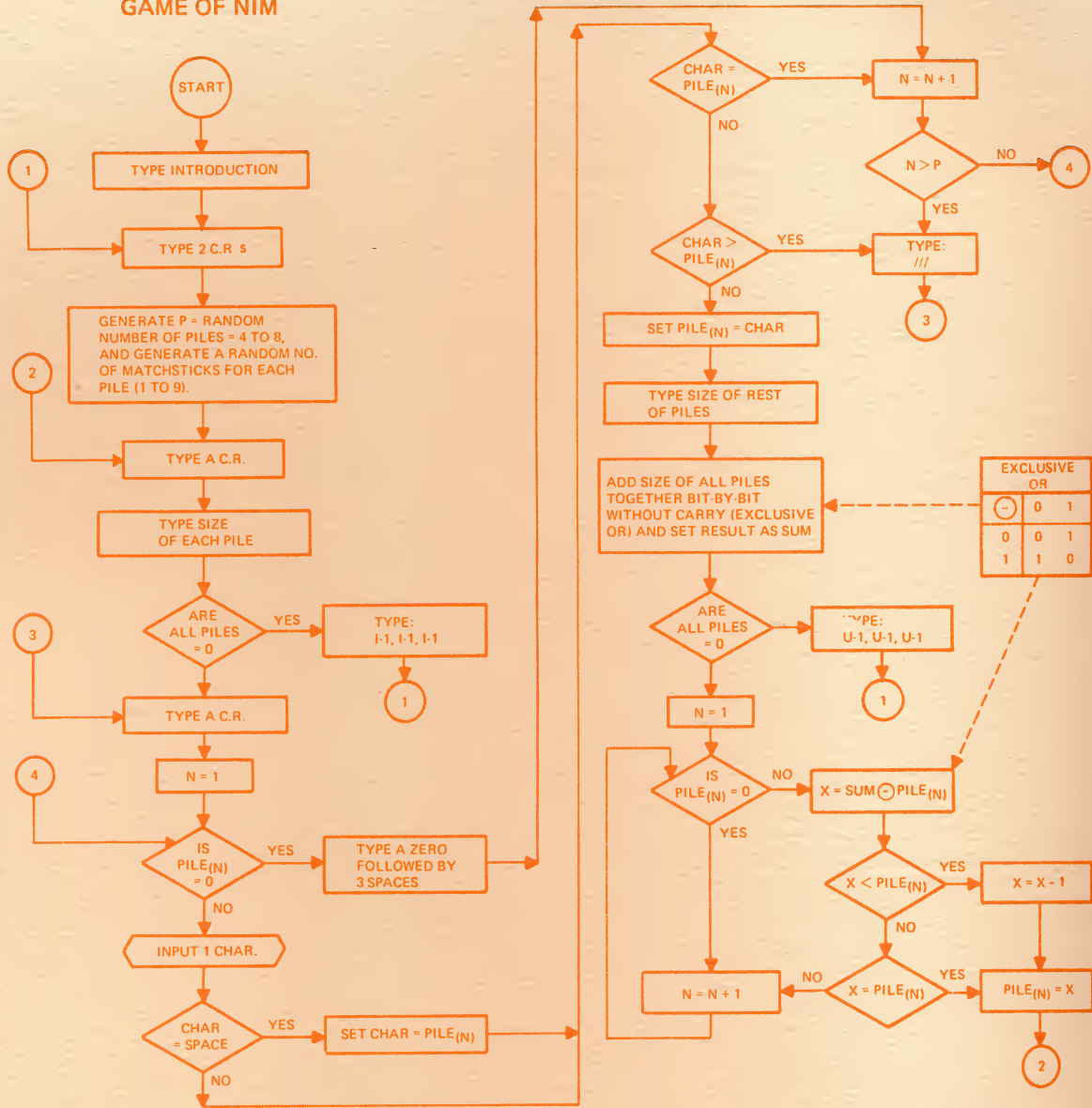
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



GAME OF NIM



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

APRIL					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
			0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0
0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1
0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0
1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1
1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0		

The game of NIM is usually played with piles of matchsticks, pebbles, paper clips, or other handy objects. The number of piles is variable and the number of objects in each pile is also variable. Each player may reduce any one pile to a smaller number of objects including zero. Usually, two players are involved and they alternate turns until the winning player takes the last object from the last pile.

The best move is determined by a binary analysis of the pile sizes as follows: Each pile size is expressed as a binary number and then these numbers are added column by column with no carry between columns (exclusive or). The object is to make this sum zero. Set X equal to this sum exclusive or'd (add without carry) with the first non-zero pile size: If X is greater than the pile size, that pile should not be reduced and this procedure is repeated on the next non-zero pile; if X is equal to the pile size (this means the sum was already zero and the other player is in the "winning mode"), the best move is to reduce that pile by 1 in order to extend the game as long as possible (giving the other player the greatest number of chances to make a wrong move). If X is less than the pile size, the best move is to set the pile size equal to X .

Pile Size (Decimal)	P = Pile Size (Binary)	X = P – S (Binary)	Action Taken
5	0101	1111	X > P; next pile
2	0010	1000	X > P; next pile
7	0111	1101	X > P; next pile
9	1001	0011	X < P; set pile to X = 3 (next move)
3	0011	1001	
S = SUM = 1010			
Fig. 1: NIM solution example			

The computer acts as one of the players, and since the first player has a $N:1$ chance of winning (where N is the largest size pile), the operator is allowed to go first. However, if the player makes one wrong move at any time, the computer will lock-in to "winning mode" and will win the game. Since humans are not very good at binary arithmetic, the chances of the operator making a wrong move is very great, while the computer will never error (its been said).

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.

© Copyright 1969

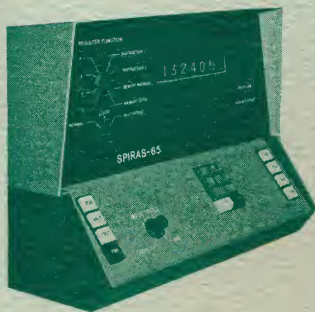
All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



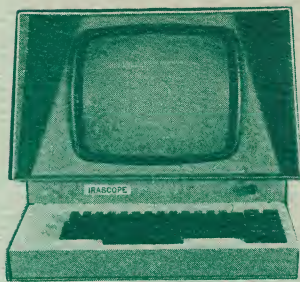
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



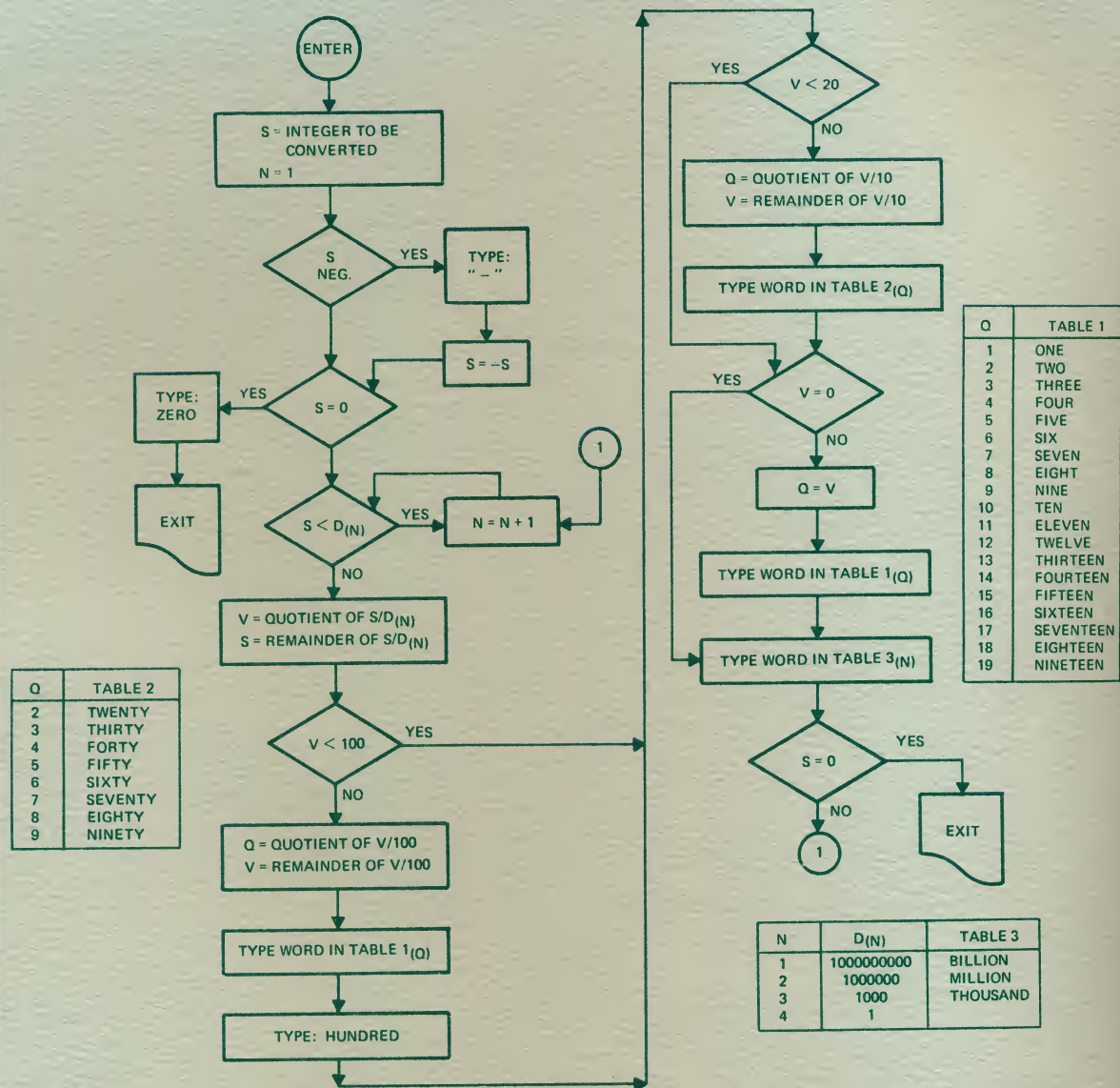
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



BINARY TO ENGLISH CONVERSION



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

MAY					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
					0 0 0 0 1	0 0 0 1 0
0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1
0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0
1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1
1 1 0 0 0 1 1 1 1 1	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0

PROGRAM OF THE MONTH: Binary to English Conversion

Numbers are usually represented by a sequence of Arabic digits. However, there are applications where an English word representation of a number is required, such as on a pay check.

This program also makes an impressive demonstration for your computer. After all, much publicity has been given to computers that can convert Russian to English or German to English. This program will convert binary to English.

Since the English language has proven itself to be illogical in spelling, definitions, etc., it is only normal that no simple relation occurs between a decimal number and its English language equivalent. It is interesting to note, however, that the conversion process between 1 to 999 repeats itself every time a decimal number has a comma. Example: 123,456,789,212 could be broken at the commas into 4 numbers — 123—456—789—212, and each one processed by the same subroutine that will convert a number between 1 and 999 to English. Each group is followed by the appropriate word, such as thousand, million, etc., to complete the conversion. Therefore, 123,456,789,212 after conversion is one hundred twenty three billion, four hundred fifty six million, seven hundred eighty nine thousand, two hundred twelve.

For converting a number into dollars and cents:

- (1) Separate the dollars from the cents (usually a division by 100 is sufficient).
- (2) Process the dollars portion as shown in the flowchart.
- (3) Type the words "DOLLARS AND."
- (4) Process the cents portion as shown in the flowchart.
- (5) Type the word "CENTS."

If all calculations are done using fixed point integer arithmetic, the conversion will be exact.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.

FOR

Industrial/Manufacturing Processes
Numerical Machine Tools
Data Acquisition and Analysis
Automatic Test and Instrumentation
Function Generation
CRT Display Graphics Generation
Communications Data Concentration and Distribution
Scientific Data Processing
Business Data Processing

© Copyright 1969

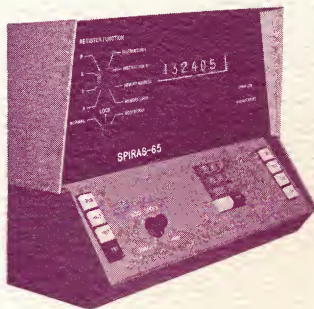
All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



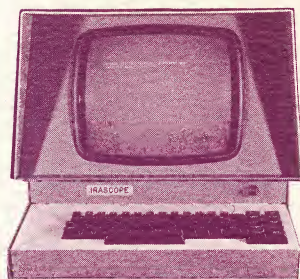
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



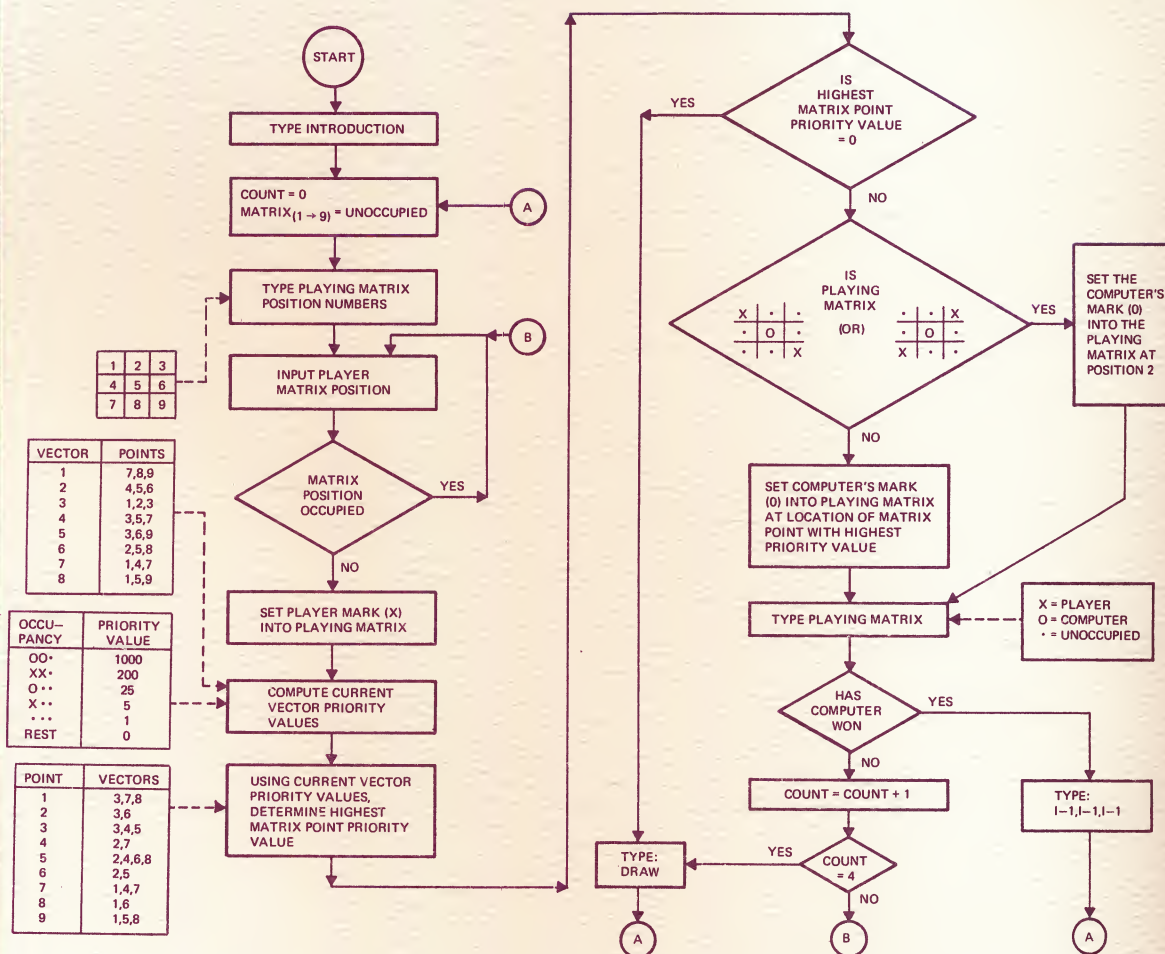
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



TIC-TAC-TOE



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

JUNE				3 6 6 2 8		
SUN	MON	TUES	WED	THUR	FRI	SAT
	0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0
0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1
0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0
1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1
1 1 1 0 0	1 1 1 0 1	1 1 1 1 0				

PROGRAM OF THE MONTH: Tic-Tac-Toe

Since a straight forward table look-up procedure for the solution of two dimensional tic-tac-toe would require 3^9 words or a 58749 word table, a more sophisticated solution procedure is used.

First, the playing matrix is organized as points 1 through 9 with 8 possible ways of winning which will be called vectors 1 through 9. (See Figure 1.)

Each vector is then analyzed and given action priorities depending on the occupancy of each vector. For instance, a vector which contains one unoccupied position and two marks for the player will be given a high priority because unless the unoccupied position is taken, the player will win. However, a vector which has an unoccupied position and two marks for the computer will be given a still higher priority because filling the unoccupied position will give the computer a win. The priority values given to each vector are shown in Figure 2.

Now, in order to determine which point in the playing matrix is the best next move, the vector values for all the vectors passing through any given point is added together to give the priority value for that point. For example, point 1 priority value would be the sum of vector 1, 4, and 7 vector priorities. Point 5 priority value would be the sum of vector 2, 4, 6, and 8 vector priorities, etc. The point with the highest priority sum determines the computer's next move. There are, however, two exceptions as shown in Figure 3. In either of these situations, the computer logic would have chosen a corner and this is a losing move. Therefore, this special situation is checked for and, when found, the computer will always take position 2.

A draw game exists whenever the highest point priority value is zero, or after the eight matrix entry where no winning move exists. If the player is skilled, all games should end in a "draw." If a player error is made, the computer will win. The computer will never lose.

In order to make the game less trivial to a skilled player, an option of presenting the player matrix in a one-dimension line is possible as shown in Figure 4. The playing rules are the same and the computer's playing logic are the same, but to the player, the winning vectors are not as obvious as they are in a two dimensional playing matrix display.

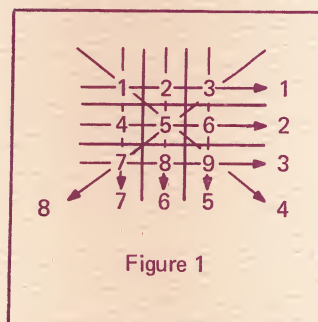


Figure 1

Vector Occupancy (or any permutation)	Priority Value
O O .	1000
X X .	200
O . .	25
X . .	5
. . .	0
O = Computer X = Player • = Unoccupied	

Figure 2

X . .	or	. . X
. O .		. O .
. . X		X . .

Figure 3

1	2	3	4	5	6	7	8	9
.	X	O	.	O	.	X	.	.
etc.								

Figure 4

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



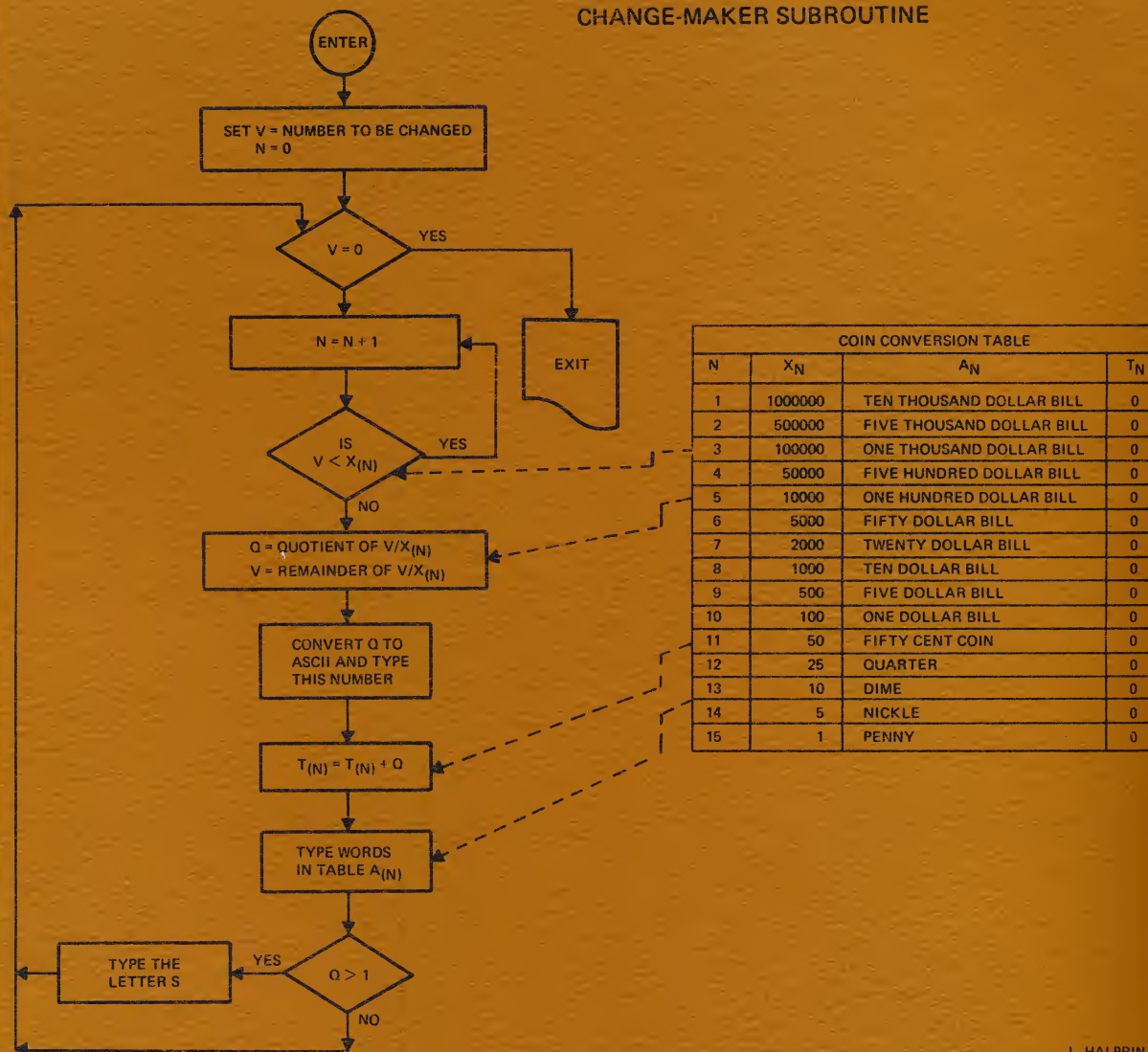
SPIRAS-65 COMPUTER



IRASCOPE
DATA TERMINAL



CHANGE-MAKER SUBROUTINE



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

JULY					3 6 6 2 g	
SUN	MON	TUES	WED	THUR	FRI	SAT
			0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0
0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1
0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0
1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1
1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1	

PROGRAM OF THE MONTH: Change Maker

Since programmers have an affinity for change (changing their flowcharts, changing their minds, changing their jobs, etc.), this procedure for making change should be appealing to them.

Such a change-maker subroutine has its uses in payroll programs where actual money must be divided into pay envelopes or for demonstrations to computer unoriented types who are still impressed by seeing a typewriter print by itself. An input value of 2468 (\$24.68), for example, would type: 1 twenty dollar bill, 4 one dollar bills, 1 fifty cent coin, 1 dime, 1 nickle, and 3 pennies.

The United States currency system includes paper bills in denominations of \$1, 2, 5, 10, 20, 50, 100, 500, 1000, 5000, and 10,000, and coins of 1, 5, 10, 25, 50, and 100. The amount to be converted to change is reduced to the number of each of these denominations needed by a successive series of tests and divisions starting with the largest denominator. The \$2 dollar bill and 100 coin are not included, due to their infrequent use in change making.

No simple mathematical equation is available for converting money to change, but the reduction method shown in the flowchart does lend itself nicely to an indexed loop, and the table used is relatively short. Also, the table can easily be modified for currency systems used by other nations. If fixed point integer arithmetic is used, the conversion to change is exact.

A coin accumulation table (T_N) is included to give the total number of each size coin or bill needed for all values converted to change, since the last time this table was initialized.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.

FOR

Industrial/Manufacturing
Processes
Numerical Machine Tools
Data Acquisition and Analysis
Automatic Test and
Instrumentation
Function Generation
CRT Display Graphics
Generation
Communications Data Con-
centration and Distribution
Scientific Data Processing
Business Data Processing

© Copyright 1969

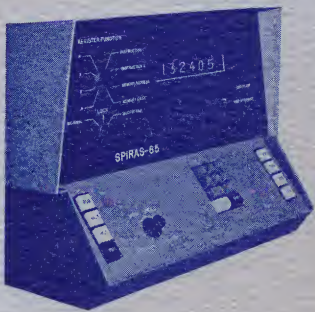
All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



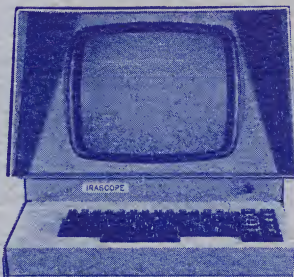
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



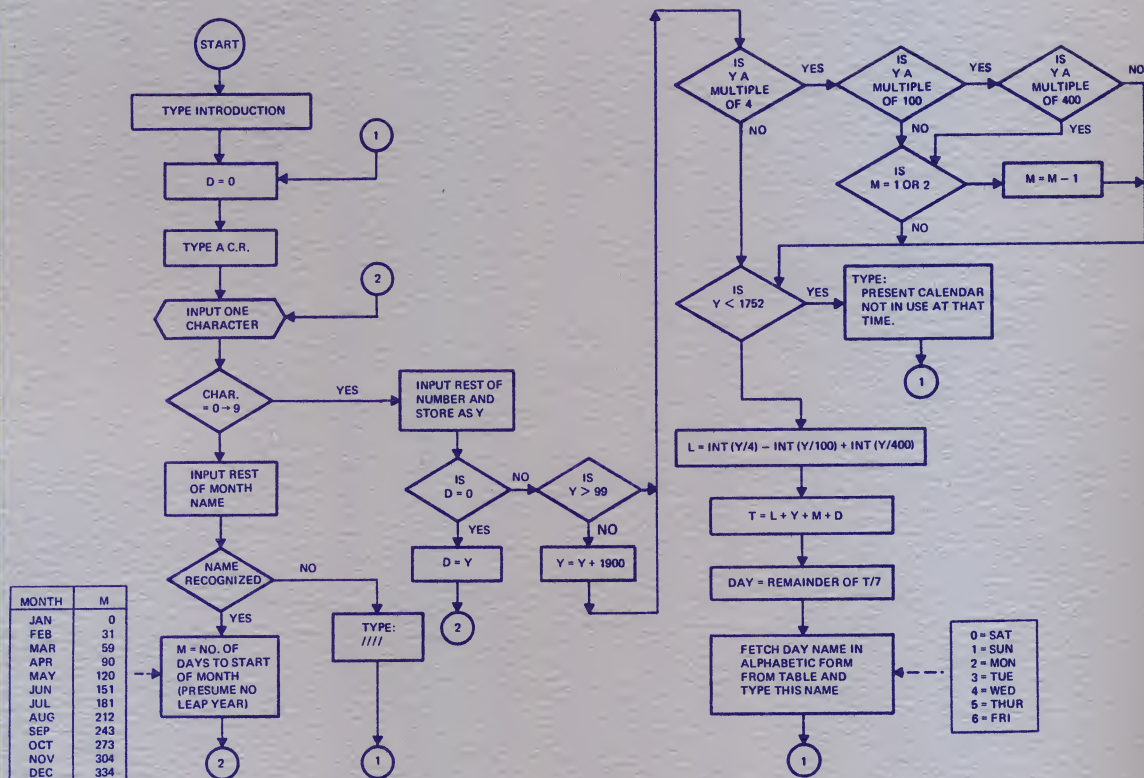
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



DATE FINDER



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

AUGUST							3 6 6 2 8
SUN	MON	TUES	WED	THUR	FRI	SAT	
						0 0 0 0 1	
0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	
0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	
1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	
1 0 1 1 1 1 1 1 1 0	1 1 0 0 0 1 1 1 1 1	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	

PROGRAM OF THE MONTH: Date Finder

Many computers have built-in clocks installed. This program will satisfy those who wish to install a built-in calendar. The computer will accept any month-day-year combination (after 1752) and tell the operator what day of the week it fell/falls on.

The Gregorian calendar now in use was adopted by England and her colonies in 1752. A year on this calendar is usually 365 days. However, in order to adjust for an inaccuracy in this figure, every year divisible by 4 (leap year) has 366 days. The year is still not quite correct, so an additional adjustment is made every century as follows: every year divisible by 100 has 365 days (even though it is a leap year) except those years divisible by 400 (leap century) which has 366 days.

The number of days from year 0 to the requested date is computed as shown in Figure 1. All arithmetic is integer.

$$\text{DAYS} = (365 * Y) + Y/4 - Y/100 + Y/400 + M + D$$

No. of Leap Years

Adjust for centuries

No. of leap centuries

No. of days from year zero to year specified.

No. of days specified

No. of days in year to start of month specified (presume no leap year)

Note: Since the remainder of $(365 * Y)/7$ is the same as the remainder of $Y/7$, the term Y could be substituted for $(365 * Y)$.

Arithmetic with no rounding of intermediate calculations. The resulting days are then divided by 7 and the remainder from this division (0 – 6) represents the day of the week on which the date fell (or will fall).

The flowchart allows several liberties. If only the first three letters of the month are used, it makes no difference if the month name is spelled out or is abbreviated. If the year is less than 100, it is presumed in the twentieth century. Depending on what characters are accepted as terminators, the following date formats are acceptable by this logic: Jan 1, 1960; January 1, 1960; Jan 1, 60; 1 Jan 60; 1/Jan/60; 1 January 1960, etc.

The following dates can be used to check-out the completed programs: Jan 4, 1756 = Sun; Feb 12, 1803 = Sat; Mar 20, 1840 = Fri; Apr 30, 1875 = Fri; May 4, 1903 = Mon; Jun 8, 1920 = Tue; Jul 15, 1942 = Wed; Aug 24, 1961 = Thur; Sept 25, 1964 = Fri; Oct 4, 1975 = Sat; Nov 17, 1985 = Sun; and Dec 25, 1999 = Sat.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

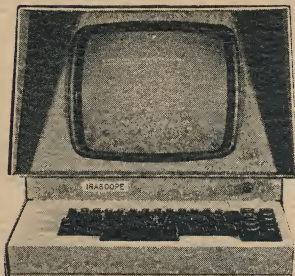
USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



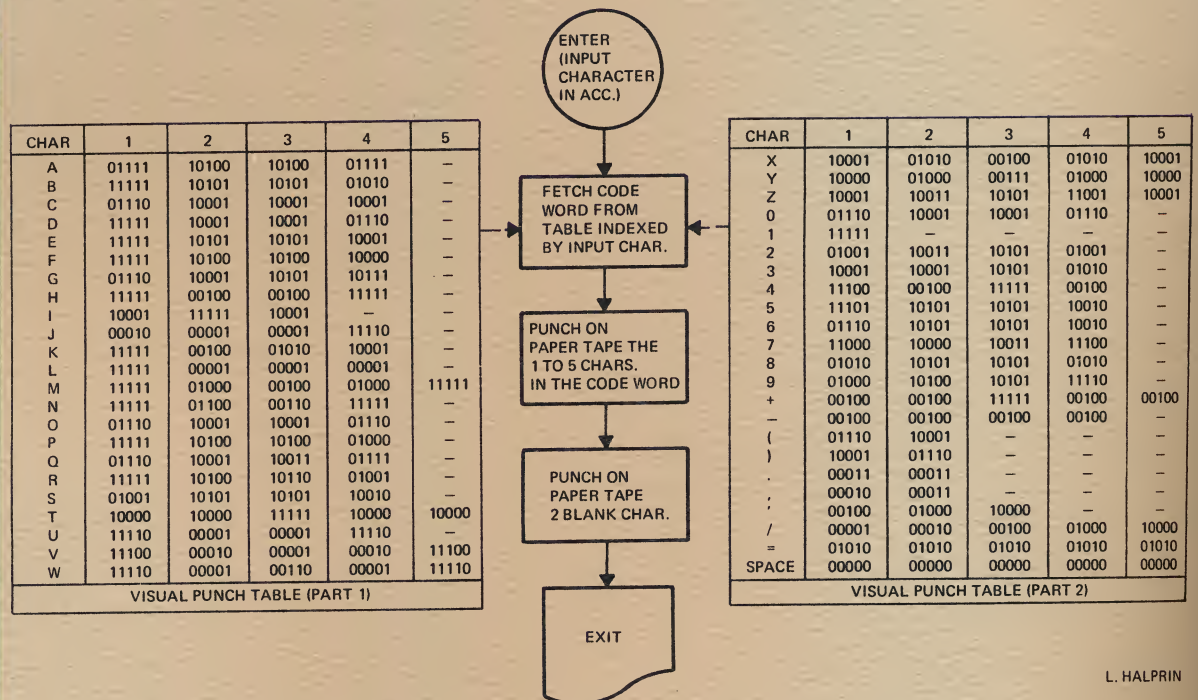
SPIRAS-65 COMPUTER



IRASCOPE
DATA TERMINAL



VISUAL PUNCH OUTPUT



SPIRAS PROGRAMMER'S CALENDAR ©

SEPTEMBER					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
		0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1
0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0
0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1
1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0
1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0			

PROGRAM OF THE MONTH: Visual Punch Output

The visual punch program converts a character in BCD form to a series of paper tape punches that, when looked at visually, form a facsimile of the BCD character. This conversion is useful when the front of tapes are to be labeled for the operator's convenience, or for punching a message (such as "END") at the end of a tape so the tape will not be loaded backwards.

The subroutine also makes the heart of a demonstration program where the operator can type his name or a message through the console keyboard and have a visual readable copy of that message punched on paper tape. (With mylar tape and appropriate messages, personalized gift wrapping ribbon can be generated.)

Each character is expressed in a variable size matrix from 1x5 to 5x5. The bits in each character may have to be rearranged depending on (1) the flexibility of the computer's paper tape punch system, (2) which edge of the tape is to be the bottom of the character, and (3) how the parity or any other control bits are arranged on the tape. Most tape systems have seven bit levels allowing 5 bits as shown in the visual punch table, an unused (buffer) bit, and the last bit is used as necessary to satisfy parity.

The following figures show the visually punched form of the characters in the visual punch table.



© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



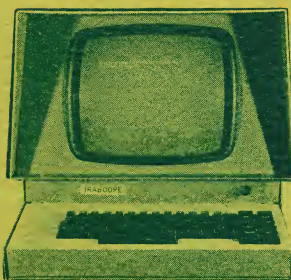
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

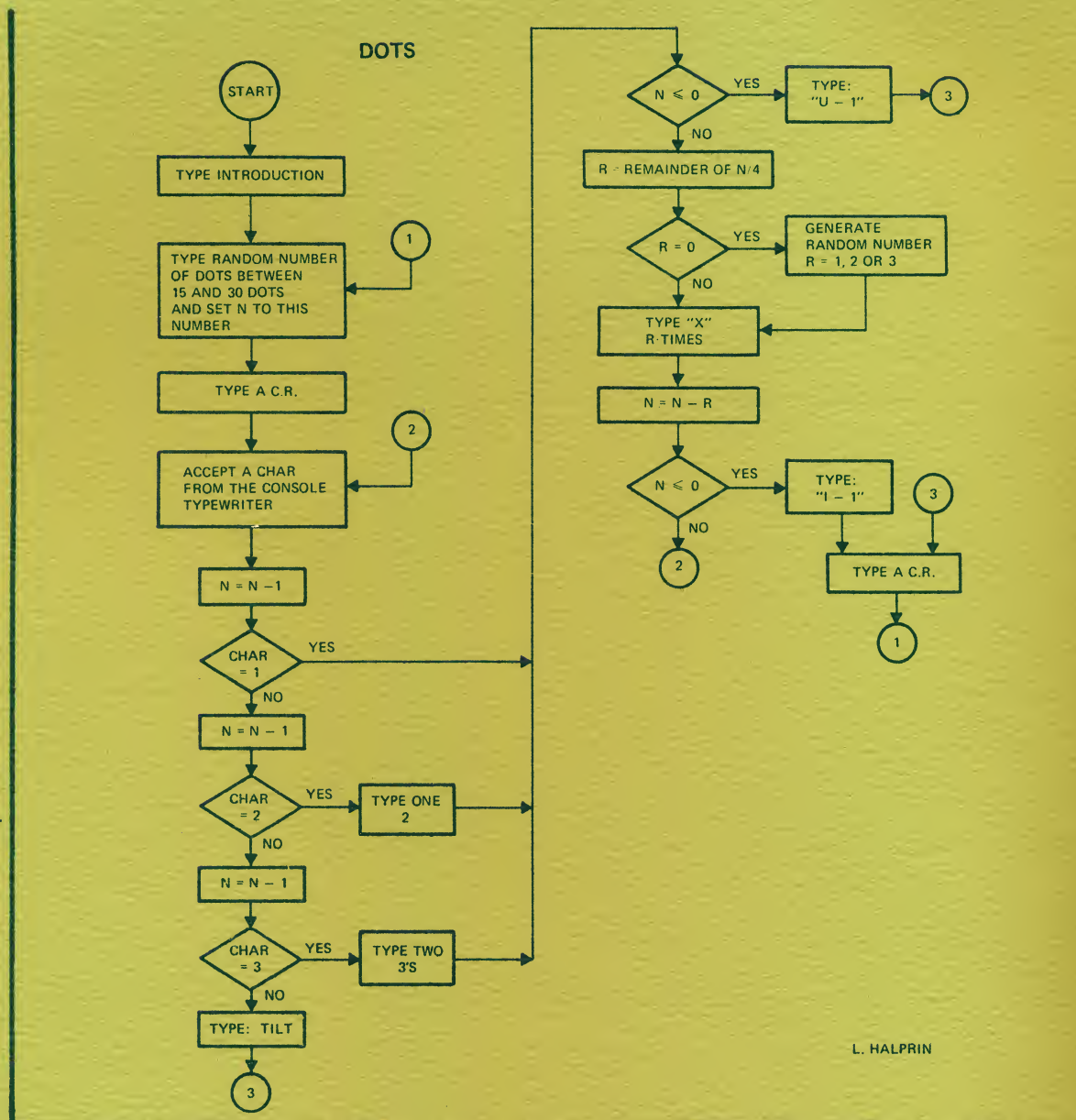
USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



SPIRAS PROGRAMMER'S CALENDAR ©

OCTOBER					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
				0 0 0 0 1	0 0 0 1 0	0 0 0 1 1
0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0
0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1
1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0
1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1

PROGRAM OF THE MONTH: Dots

The game of DOTS is a variation and simplification of the game of NIM. It is usually played with a single but variable size pile of objects (usually matchsticks). Each of two players alternately remove 1, 2, or 3 of the objects until the winning player removes the last object.

When the pile contains only 4 objects, the player whose turn it is can remove 1, 2, or 3 objects, leaving 3, 2, or 1 object. The other player need only remove those objects left in order to win. Therefore, the player who is left with only 4 objects in the pile cannot win (unless the other player makes an error). The same logic holds if the pile contains 8 objects, since in one play alternation the other player could always reduce the pile to 4 objects and, in the next play, to zero objects and win. The winning strategy, then, is to always reduce the number of objects in the pile to a multiple of 4.

The computer acts as one of the players and starts by typing a line of dots (representing the pile) which varies randomly between games from 15 to 30 dots long. The operator then types a 1, 2, or 3, presenting the number of dots he wishes to remove from the pile. The computer will attempt to reduce the pile size to a multiple of 4 and will type X's representing the number of dots it removed from the pile. This procedure alternates until the last dot is taken and the winner declared.

Notice, the operator could win for any size pile of dots (between 15 and 30), except 16, 20, 24, or 28. This seems to give a 4:1 advantage to the player, but parallax and blurring present in a long string of dots can easily give the operator a wrong count when the size of the pile is hastily counted. Also, even if the operator does count correctly, if he makes just one error during the game, the computer will immediately latch on to the winning mode. The computer, of course, will never make a mistake.

Alternate presentations of this game on a computer could be a programmable register's binary console display for showing the current size of the pile (bits instead of dots) and 3 sense switches could provide the operator input. Those computers with Nixie type displays or oscilloscope display tubes on their console lend themselves for interesting game presentations.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



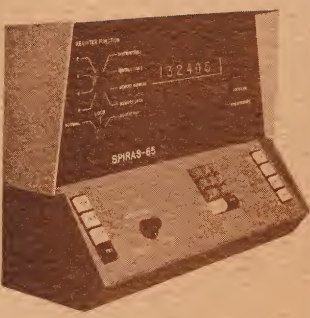
Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.

© Copyright 1969

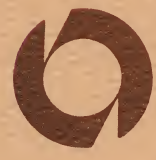
All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.



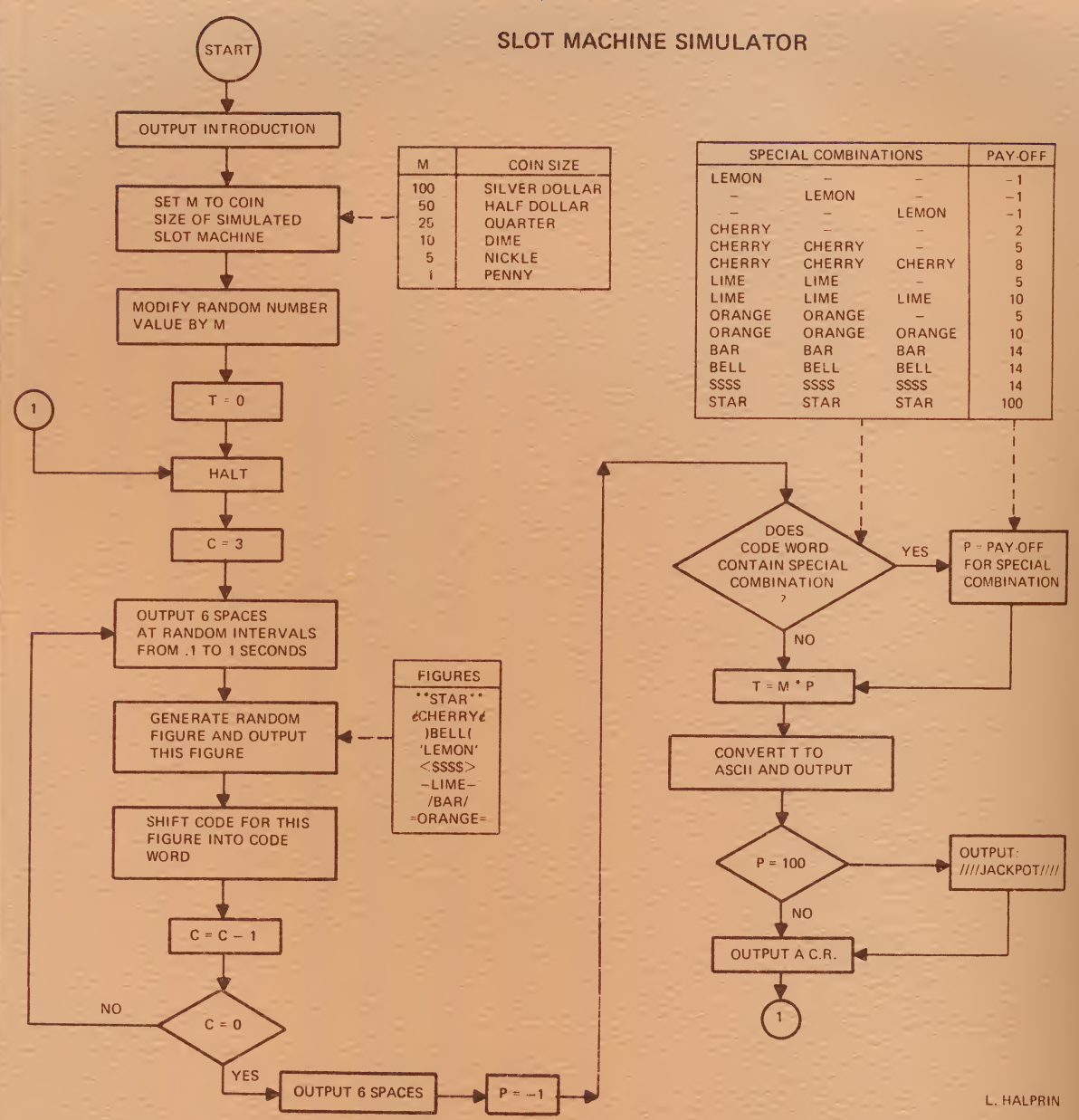
SPIRAS-65 COMPUTER



IRASCOPE DATA TERMINAL



SLOT MACHINE SIMULATOR



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

NOVEMBER					3 6 6 2 8	
SUN	MON	TUES	WED	THUR	FRI	SAT
0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1	0 0 1 1 0	0 0 1 1 1
0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0	0 1 1 0 1	0 1 1 1 0
0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1	1 0 1 0 0	1 0 1 0 1
1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0	1 1 0 1 1	1 1 1 0 0
1 1 1 0 1	1 1 1 1 0					

PROGRAM OF THE MONTH: Slot Machine Simulator

There are many people who consider the use of a computer as being a big gamble. For those who literally wish to gamble with their computer, this slot machine simulator may provide the means.

The console typewriter is used as the "box", the computer's start button is the "handle", and, instead of figures showing behind the "windows", the typewriter instead types three words representing the figures. There are 8 "figures" and the pay-offs vary approximately the same as the "slots" at Las Vegas (see table on flow charts).

The coin size of the simulator may be set by sense switches, initial key words, or by other methods. The "figures" are chosen in a random order, but an observant operator might soon notice that the sequence of figures is identical every time the program is re-loaded into the computer. This is due to the fact that although a random number "generator" function will give fairly good random numbers, the sequence will always be identical whenever the function is set to the same starting value (such as occurs when the program is loaded into memory). To minimize this problem, the initial value of the random number generator is modified by the coin size chosen for the slot machine simulator. Since this coin size is a variable, the "figure" sequence is much less obvious.

A running total of money won or lost by the operator is accumulated and typed after each "pull." No money is actually passed between the computer and the operator, but the imaginative engineer could easily rig up a coin slot and handle to the computer's in/out or real time interrupt logic for a more realistic (and probably illegal) slot machine simulator.

The simulator could easily be extended to a four window slot machine simulator and more "figures" could be added as desired. A computer with a console oscilloscope display tube and a clever programmer could show actual figures revolving behind "windows." Those computers with programmable nixie console displays could replace the figures with a digit 0 to 7 and then 'revolve' these figures instead. As you can see, this demonstration gives the programmer a lot of latitude in display formats.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65K of memory, representing unprecedented PERFORMANCE PER DOLLAR.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.

FOR

Industrial/Manufacturing Processes
Numerical Machine Tools
Data Acquisition and Analysis
Automatic Test and Instrumentation
Function Generation
CRT Display Graphics Generation
Communications Data Concentration and Distribution
Scientific Data Processing
Business Data Processing

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

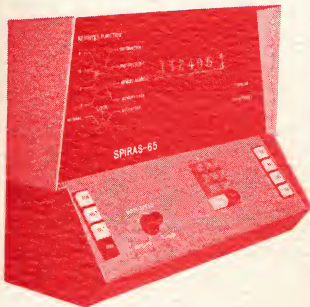


Spiras Systems, Inc.

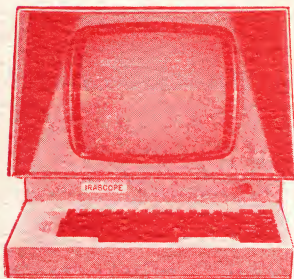
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.



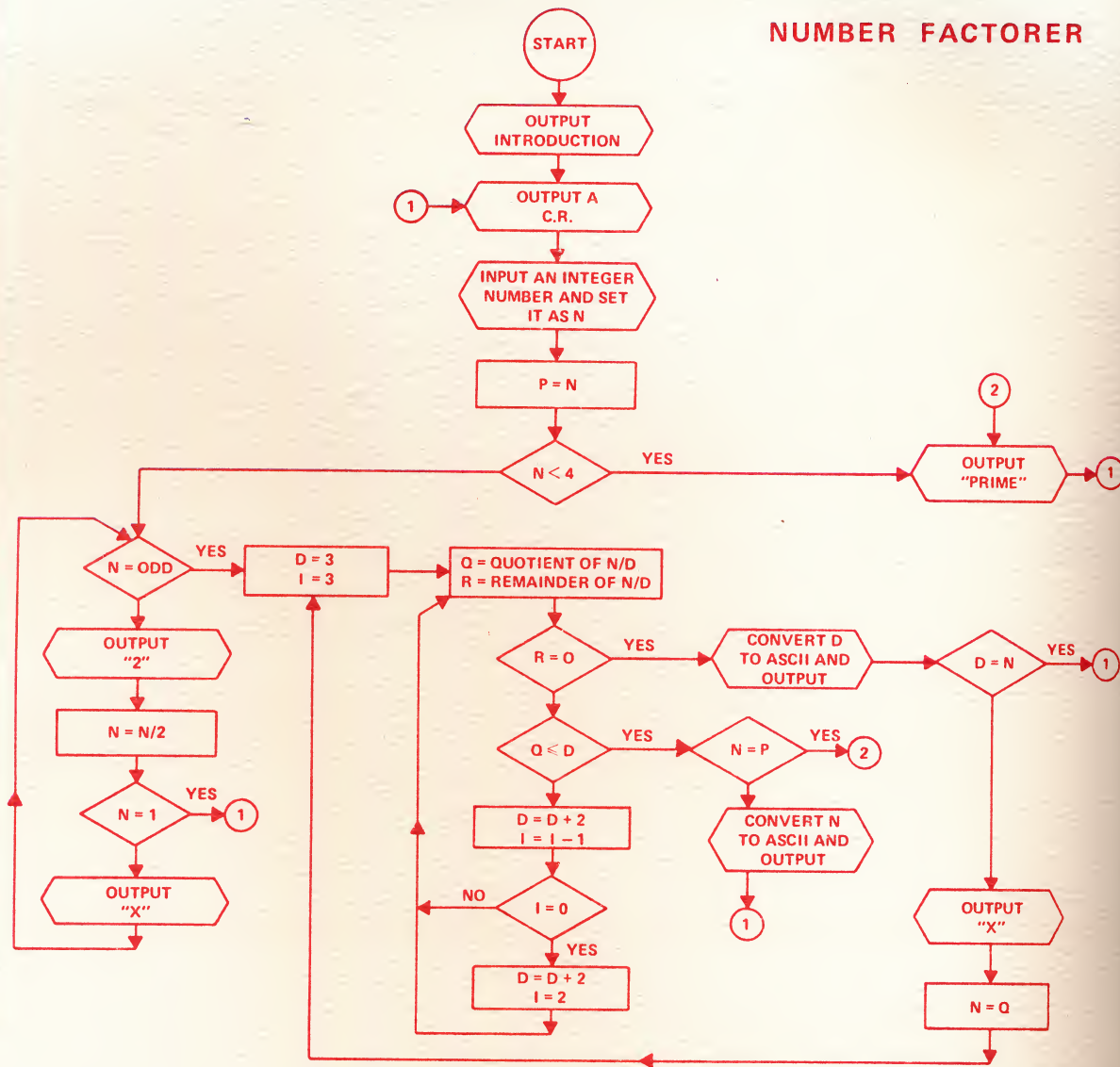
SPIRAS-65 COMPUTER



IRASCOPE
DATA TERMINAL



NUMBER FACTORER



L. HALPRIN

SPIRAS PROGRAMMER'S CALENDAR ©

DECEMBER						3 6 6 2 8
SUN	MON	TUES	WED	THUR	FRI	SAT
		0 0 0 0 1	0 0 0 1 0	0 0 0 1 1	0 0 1 0 0	0 0 1 0 1
0 0 1 1 0	0 0 1 1 1	0 1 0 0 0	0 1 0 0 1	0 1 0 1 0	0 1 0 1 1	0 1 1 0 0
0 1 1 0 1	0 1 1 1 0	0 1 1 1 1	1 0 0 0 0	1 0 0 0 1	1 0 0 1 0	1 0 0 1 1
1 0 1 0 0	1 0 1 0 1	1 0 1 1 0	1 0 1 1 1	1 1 0 0 0	1 1 0 0 1	1 1 0 1 0
1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1		

PROGRAM OF THE MONTH: Number Factorer

An old stand-by in computer demonstrations is a number factoring program in which an integer is input from the operator and the prime factors of that number are typed by the computer.

The number is converted to its prime factors by the following logic:

- (1) If the number is less than 4, it is prime.
- (2) If the number is even, a 2 is typed (one of its factors), the number is divided by 2, and this step is repeated until the number becomes 1 (conversion complete) or until the number becomes odd (go to step 3).
- (3) If the number is odd, it is divided by a sequence of semi-prime numbers (d) where $D = 3, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31$, etc.
- (4) Whenever the remainder of one of these divisions is zero, type that value of D (as one of the number's prime factors). If the quotient of this division is 1, the conversion is complete. Otherwise, set the quotient as the new number, reset D to 3, and repeat steps 3, 4, and 5.
- (5) Whenever the quotient resulting from a division during step 3 becomes equal or less than D, then that number is the last prime factor of the original number. If that number is the original number, type the word "prime." Otherwise, type that number. In either case, the conversion is completed.

An automatic demonstration can be had by programming a sequential counter starting at 1. The contents of this counter are typed, followed by its prime factors, and then the counter is incremented. A variation of this procedure would replace the sequential counter with a random number generator set to generate random integers within a reasonable range.

SPIRAS-65 is a General Purpose Stored Program Controller/Processor expandable to 65 K of memory, representing unprecedented **PERFORMANCE PER DOLLAR**.

SPIRAS-65 may be efficiently applied as a Data Processor or Real-Time Controller in either a dedicated and/or satellite environment.



Spiras Systems, Inc.
Affiliate of
USM Corporation

332 Second Avenue
Waltham, Massachusetts 02154
617-891-7300 TWX 710-324-6699

USM is a world-wide 400 million dollar designer and manufacturer of systems, aimed at improving the productivity of manufacturing and service industries.

© Copyright 1969

All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.